Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2422vs020eg

Table of Contents

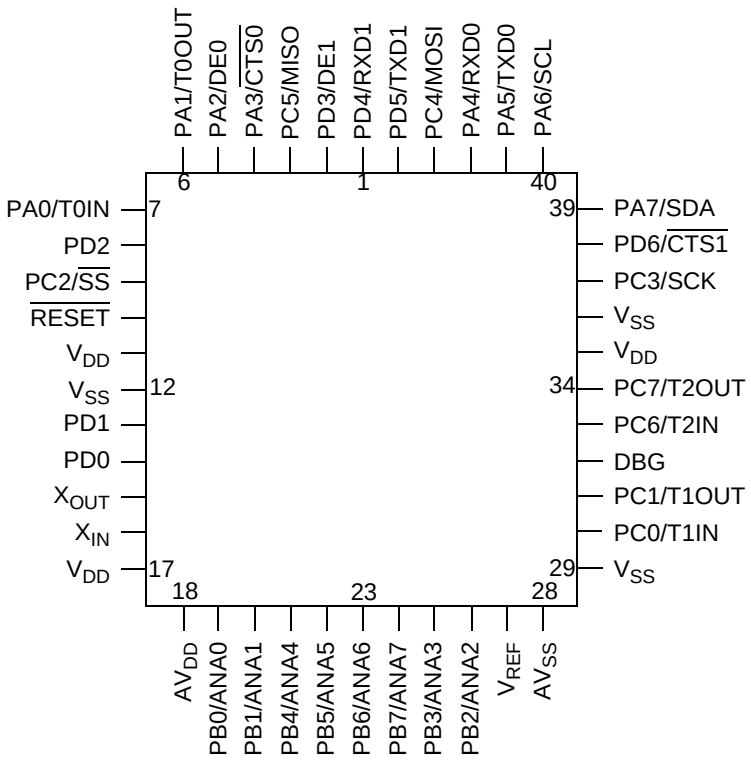

Figure 3. Z8 Encore! XP F64xx Series in 44-Pin Plastic Leaded Chip Carrier (PLCC)

Note: Timer 3 is not available in the 44-pin PLCC package.

Voltage Brown-Out Reset

The devices in the Z8 Encore! XP F64xx Series provide low Voltage Brown-Out protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage remains below the Power-On Reset voltage threshold (V_{POR}), the VBO block holds the device in the Reset state.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the devices progress through a full system reset sequence, as described in the Power-On Reset section. Following Power-On Reset, the POR status bit in the Watchdog Timer Control (WDTCTL) Register is set to 1. Figure 9 displays Voltage Brown-Out operation. For the VBO and POR threshold voltages (V_{VBO} and V_{POR}), see the Electrical Characteristics chapter on page 200.

The Voltage Brown-Out circuit can be either enabled or disabled during STOP Mode. Operation during STOP Mode is set by the VBO_AO option bit. For information about configuring VBO_AO, see the Option Bits chapter on page 180.

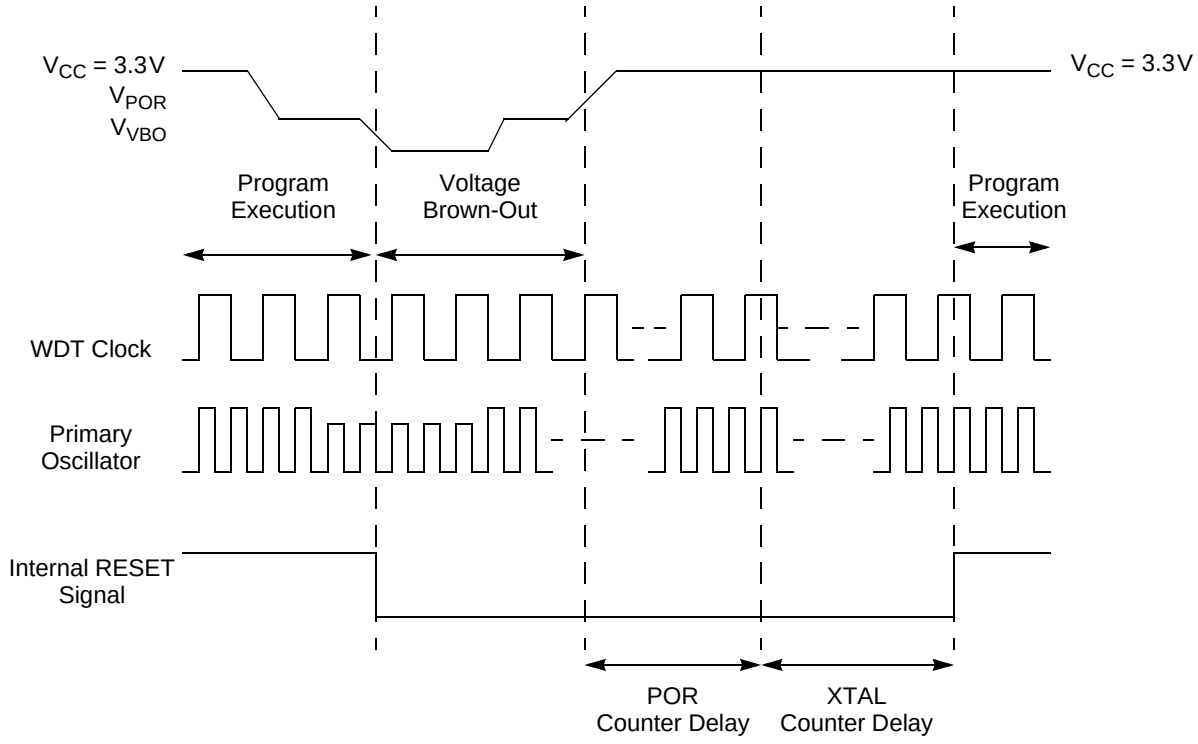

Figure 9. Voltage Brown-Out Reset Operation

Table 10. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
STOP Mode	Watchdog Timer time-out when configured for Reset.	Stop Mode Recovery.
	Watchdog Timer time-out when configured for interrupt.	Stop Mode Recovery followed by interrupt (if interrupts are enabled).
	Data transition on any GPIO port pin enabled as a Stop Mode Recovery source.	Stop Mode Recovery.

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control Register, the WDT and stop bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and the Z8 Encore! XP F64xx Series devices are configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery Using a GPIO Port Pin Transition HALT

Each of the GPIO port pins may be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. The GPIO Stop Mode Recovery signals are filtered to reject pulses less than 10 ns (typical) in duration. In the Watchdog Timer Control Register, the stop bit is set to 1.

! **Caution:** In STOP Mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin through the end of the Stop Mode Recovery delay. Thus, short pulses on the Port pin can initiate Stop Mode Recovery without being written to the Port Input Data Register or without initiating an interrupt (if enabled for that pin).

Table 12. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description
Port F	PF[7:0]	N/A	No alternate functions
Port G	PG[7:0]	N/A	No alternate functions
Port H	PH0	ANA8	ADC analog input 8
	PH1	ANA9	ADC analog input 9
	PH2	ANA10	ADC analog input 10
	PH3	ANA11	ADC analog input 11

GPIO Interrupts

Many of the GPIO port pins can be used as interrupt sources. Some port pins may be configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port pin interrupts generate an interrupt when any edge occurs (both rising and falling). For more information about interrupts using the GPIO pins, see the Interrupt Controller chapter on page 47.

GPIO Control Register Definitions

Four registers for each Port provide access to GPIO control, input data, and output data. Table 13 lists these Port registers. Use the Port A–H Address and Control registers together to provide access to subregisters for Port configuration and control.

Table 13. GPIO Port Registers and Subregisters

Port Register Mnemonic	Port Register Name
P*x*ADDR	Port A–H Address Register (selects subregisters)
P*x*CTL	Port A–H Control Register (provides access to subregisters)
P*x*IN	Port A–H Input Data Register
P*x*OUT	Port A–H Output Data Register
Port Subregister Mnemonic	**Port Register Name**
P*x*DD	Data Direction
P*x*AF	Alternate Function
P*x*OC	Output Control (Open-Drain)
PxDD	High Drive Enable
P*x*SMRE	Stop Mode Recovery Source Enable

Table 23. Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt Source
Highest	0002H	Reset (not an interrupt)
	0004H	Watchdog Timer (see the Watchdog Timer chapter on page 80)
	0006H	Illegal Instruction Trap (not an interrupt)
	0008H	Timer 2
	000AH	Timer 1
	000CH	Timer 0
	000EH	UART 0 receiver
	0010H	UART 0 transmitter
	0012H	I^2C
	0014H	SPI
	0016H	ADC
	0018H	Port A7 or Port D7, rising or falling input edge
	001AH	Port A6 or Port D6, rising or falling input edge
	001CH	Port A5 or Port D5, rising or falling input edge
	001EH	Port A4 or Port D4, rising or falling input edge
	0020H	Port A3 or Port D3, rising or falling input edge
	0022H	Port A2 or Port D2, rising or falling input edge
	0024H	Port A1 or Port D1, rising or falling input edge
	0026H	Port A0 or Port D0, rising or falling input edge
	0028H	Timer 3 (not available in the 44-pin package)
	002AH	UART 1 receiver
	002CH	UART 1 transmitter
	002EH	DMA
	0030H	Port C3, both input edges
	0032H	Port C2, both input edges
	0034H	Port C1, both input edges
Lowest	0036H	Port C0, both input edges

Interrupt Control Register

The Interrupt Control (IRQCTL) Register, shown in Table 38, contains the master enable bit for all interrupts.

Table 38. Interrupt Control Register (IRQCTL)

Bit	7	6	5	4	3	2	1	0
Field	IRQE	Reserved						
RESET	0							
R/W	R/W	R						
Address	FCFH							

Bit	Description
[7] IRQE	**Interrupt Request Enable** This bit is set to 1 by execution of an EI or IRET instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, or a Reset. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	**Reserved** These pins are reserved and must be programmed to 000000.

Table 53. UART Transmit Data Register (U*x*TXD)

Bit	7	6	5	4	3	2	1	0
Field	TXD							
RESET	X							
R/W	W							
Address	F40H and F48H							

Bit	Description
[7:0] TXD	**Transmit Data** UART transmitter data byte to be shifted out through the TXD*x* pin.

UART Receive Data Register

Data bytes received through the RXD*x* pin are stored in the UART Receive Data Register, shown in Table 54. The read-only UART Receive Data Register shares a Register File address with the write-only UART Transmit Data Register.

Table 54. UART Receive Data Register (U*x*RXD)

Bit	7	6	5	4	3	2	1	0
Field	RXD							
RESET	X							
R/W	R							
Address	F40H and F48H							

Bit	Description
[7:0] RXD	**Receive Data** UART receiver data byte from the RXD*x* pin.

Slave Operation

The SPI block is configured for SLAVE Mode operation by setting the SPIEN bit to 1 and the MMEN bit to 0 in the SPICTL Register and setting the SSIO bit to 0 in the SPIMODE Register. The IRQE, PHASE, CLKPOL, WOR bits in the SPICTL Register and the NUMBITS field in the SPIMODE Register must be set to be consistent with the other SPI devices. The STR bit in the SPICTL Register may be used if appropriate to force a *start-up* interrupt. The BIRQ bit in the SPICTL Register and the SSV bit in the SPIMODE Register are not used in SLAVE Mode. The SPI baud rate generator is not used in SLAVE Mode so the SPIBRH and SPIBRL registers need not be initialized.

If the slave has data to send to the master, the data must be written to the SPIDAT Register before the transaction starts (first edge of SCK when $\overline{SS}$ is asserted). If the SPIDAT Register is not written prior to the slave transaction, the MISO pin outputs whatever value is currently in the SPIDAT Register.

Due to the delay resulting from synchronization of the SPI input signals to the internal system clock, the maximum SPICLK baud rate that can be supported in SLAVE Mode is the system clock frequency (X_{IN}) divided by 8. This rate is controlled by the SPI master.

Error Detection

The SPI contains error detection logic to support SPI communication protocols and recognize when communication errors have occurred. The SPI Status Register indicates when a data transmission error has been detected.

Overrun (Write Collision)

An overrun error (write collision) indicates that a write to the SPI Data Register was attempted while a data transfer was in progress (in either MASTER or SLAVE modes). An overrun sets the OVR bit in the SPI Status Register to 1. Writing a 1 to OVR clears this error flag. The data register is not altered when a write occurs while data transfer is in progress.

Mode Fault (Multimaster Collision)

A mode fault indicates when more than one Master is trying to communicate at the same time (a multimaster collision). The mode fault is detected when the enabled Master's $\overline{SS}$ pin is asserted. A mode fault sets the COL bit in the SPI Status Register to 1. Writing a 1 to COL clears this error flag.

Slave Mode Abort

In the SLAVE Mode of operation, if the $\overline{SS}$ pin deasserts before all bits in a character have been transferred, the transaction is aborted. When this condition occurs, the ABT bit is set in the SPISTAT Register as well as the IRQ bit (indicating the transaction is complete).

The next time $\overline{SS}$ asserts, the MISO pin outputs SPIDAT[7], regardless of where the previous transaction left off. Writing a 1 to ABT clears this error flag.

SPI Interrupts

When SPI interrupts are enabled, the SPI generates an interrupt after character transmission/reception completes in both MASTER and SLAVE modes. A character can be defined to be 1 through 8 bits by the NUMBITS field in the SPI Mode Register. In slave mode it is not necessary for $\overline{SS}$ to deassert between characters to generate the interrupt. The SPI in Slave mode can also generate an interrupt if the $\overline{SS}$ signal deasserts prior to transfer of all the bits in a character (see description of slave abort error above). Writing a 1 to the IRQ bit in the SPI Status Register clears the pending SPI interrupt request. The IRQ bit must be cleared to 0 by the Interrupt Service Routine to generate future interrupts. To start the transfer process, an SPI interrupt may be forced by software writing a 1 to the STR bit in the SPICTL Register.

If the SPI is disabled, an SPI interrupt can be generated by a Baud Rate Generator time-out. This timer function must be enabled by setting the BIRQ bit in the SPICTL Register. This Baud Rate Generator time-out does not set the IRQ bit in the SPISTAT Register, just the SPI interrupt bit in the interrupt controller.

SPI Baud Rate Generator

In SPI Master Mode, the Baud Rate Generator creates a lower frequency serial clock (SCK) for data transmission synchronization between the Master and the external Slave. The input to the Baud Rate Generator is the system clock. The SPI Baud Rate High and Low Byte registers combine to form a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator. The SPI baud rate is calculated using the following equation:

$$\text{SPI Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{2 \times \text{BRG[15:0]}}$$

Minimum baud rate is obtained by setting BRG[15:0] to 0000H for a clock divisor value of (2 X 65536 = 131072).

When the SPI is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. Observe the following procedure to configure the Baud Rate Generator as a timer with interrupt on time-out:

1. Disable the SPI by clearing the SPIEN bit in the SPI Control Register to 0.
2. Load the appropriate 16-bit count value into the SPI Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BIRQ bit in the SPI Control Register to 1.

16. The I2C Controller completes transmission of the data on the SDA signal.
17. The slave may either Acknowledge or Not Acknowledge the last byte. Because either the stop or start bit is already set, the NCKI interrupt does not occur.
18. The I2C Controller sends the stop (or RESTART) condition to the I2C bus. The stop or start bit is cleared.

Address Only Transaction with a 10-bit Address

In the situation where software wants to determine if a slave with a 10-bit address is responding without sending or receiving data, a transaction can be done which only consists of an address phase. Figure 30 displays this *address only* transaction to determine if a slave with 10-bit address will acknowledge. As an example, this transaction can be used after a write has been performed to an EEPROM to determine when the EEPROM completes its internal write operation and is again responding to I2C transactions. If the slave does not Acknowledge the transaction can be repeated until the slave is able to Acknowledge.

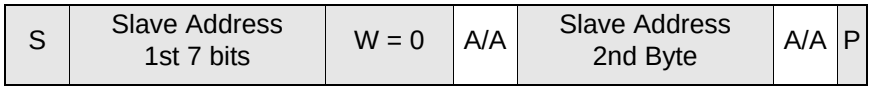

Figure 30. 10-Bit Address Only Transaction Format

Observe the following procedure for an address only transaction to a 10-bit addressed slave:

1. Software asserts the IEN bit in the I2C Control Register.
2. Software asserts the TXI bit of the I2C Control Register to enable transmit interrupts.
3. The I2C interrupt asserts, because the I2C Data Register is empty (TDRE = 1)
4. Software responds to the TDRE interrupt by writing the first slave address byte. The least significant bit must be 0 for the write operation.
5. Software asserts the start bit of the I2C Control Register.
6. The I2C Controller sends the start condition to the I2C slave.
7. The I2C Controller loads the I2C Shift Register with the contents of the I2C Data Register.
8. After one bit of address is shifted out by the SDA signal, the transmit interrupt is asserted.
9. Software responds by writing the second byte of address into the contents of the I2C Data Register.

Analog-to-Digital Converter

The Analog-to-Digital Converter (ADC) converts an analog input signal to a 10-bit binary number. The features of the sigma-delta ADC include:

- 12 analog input sources are multiplexed with general-purpose I/O ports
- Interrupt upon completion of conversion
- Internal voltage reference generator
- A Direct Memory Access (DMA) controller that can automatically initiate data conversion and transfer the data from 1 to 12 analog inputs

Architecture

Figure 34 displays the three major functional blocks (converter, analog multiplexer, and voltage reference generator) of the ADC. The ADC converts an analog input signal to its digital representation. The 12-input analog multiplexer selects one of the 12 analog input sources. The ADC requires an input reference voltage for the conversion. The voltage reference for the conversion may be input through the external V_{REF} pin or generated internally by the voltage reference generator.

Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers, shown in Tables 97 and 98, combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit Flash Frequency registers must be written with the system clock frequency in kHz for Program and Erase operations. Calculate the Flash Frequency value using the following equation:

$$FFREQ[15:0] = \{FFREQH[7:0],FFREQL[7:0]\} = \frac{\text{System Clock Frequency}}{1000}$$

! **Caution:** Flash programming and erasure is not supported for system clock frequencies below 20kHz, above 20MHz, or outside of the valid operating frequency range for the device. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper program and erase times.

Table 97. Flash Frequency High Byte Register (FFREQH)

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
RESET	0							
R/W	R/W							
Address	FFAH							

Table 98. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Bit	Description
[7:0] FFREQH, FFREQL	**Flash Frequency High and Low Bytes** These 2 bytes, {FFREQH[7:0], FFREQL[7:0]}, contain the 16-bit Flash Frequency value.

Bit	Description (Continued)
[1]	**Reserved** This bit is reserved and must be programmed to 0.
[0] FWP	**Flash Write Protect (Flash version only)** 0 = Programming, Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available through the On-Chip Debugger. 1 = Programming, and Page Erase are enabled for all of Flash program memory.

Flash Memory Address 0001H

Table 100. Options Bits at Flash Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U							
R/W	R/W							
Address	Program Memory 0001H							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7:0]	**Reserved** These option bits are reserved for future use and must always be 1. This setting is the default for unprogrammed (erased) Flash.

Bit	Description (Continued)
[4] BRKLOOP	**Breakpoint Loop** This bit determines what action the OCD takes when a BRK instruction is decoded if breakpoints are enabled (BRKEN is 1). If this bit is 0, then the DBGMODE bit is automatically set to 1 and the OCD entered DEBUG Mode. If BRKLOOP is set to 1, then the eZ8 CPU loops on the BRK instruction. 0 = BRK instruction sets DBGMODE to 1. 1 = eZ8 CPU loops on BRK instruction.
[3:1]	**Reserved** These bits are reserved and must be programmed to 000.
[0] RST	**Reset** Setting this bit to 1 resets the Z8 Encore! XP F64xx Series devices. The devices go through a normal Power-On Reset sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 when the reset finishes. 0 = No effect. 1 = Reset the Z8 Encore! XP F64xx Series device.

OCD Status Register

The OCD Status Register, shown in Table 104, reports status information about the current state of the debugger and the system.

Table 104. OCD Status Register (OCDSTAT)

Bit	7	6	5	4	3	2	1	0
Field	IDLE	HALT	RPEN	Reserved				
RESET	0							
R/W	R							

Bit	Description
[7] IDLE	**CPU Idle** This bit is set if the part is in DEBUG Mode (DBGMODE is 1), or if a BRK instruction occurred since the last time OCDCTL was written. This can be used to determine if the CPU is running or if it is idling. 0 = The eZ8 CPU is running. 1 = The eZ8 CPU is either stopped or looping on a BRK instruction.
[6] HALT	**HALT Mode** 0 = The device is not in HALT Mode. 1 = The device is in HALT Mode.

Figure 46 displays the maximum HALT Mode current consumption across the full operating temperature range of the device and plotted opposite the system clock frequency. All GPIO pins are configured as outputs and driven High.

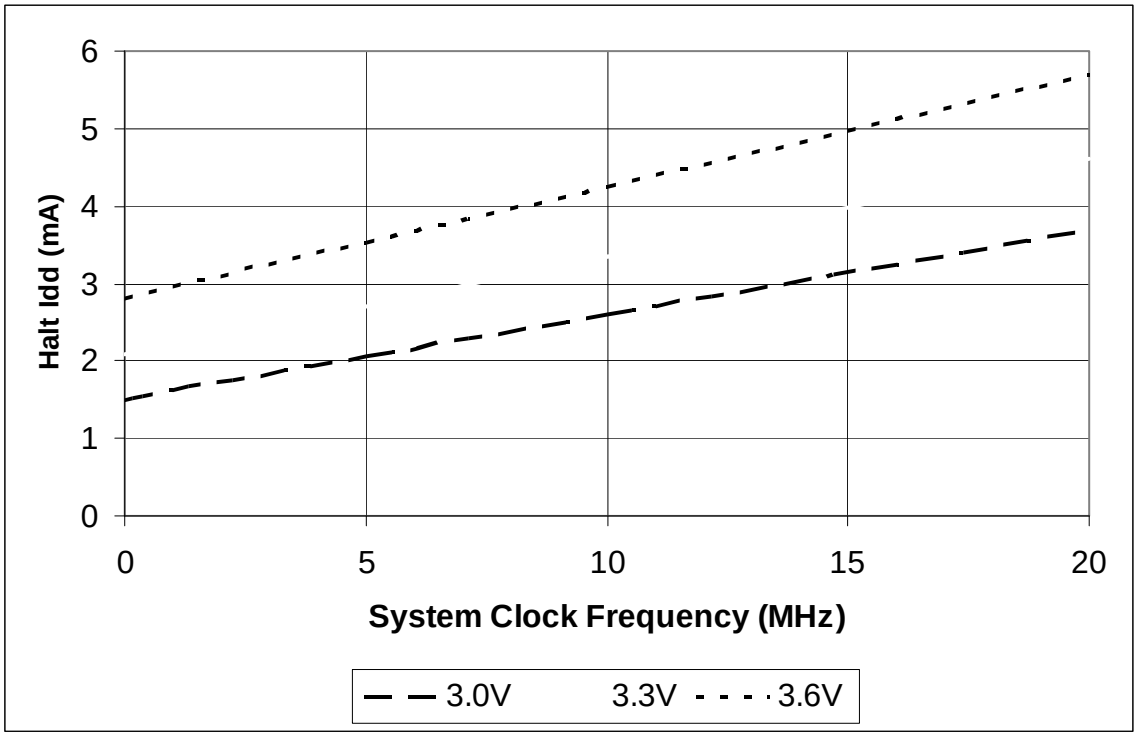

Figure 46. Maximum HALT Mode I_{CC} vs. System Clock Frequency

Table 125. Notational Shorthand (Continued)

Notation	Description	Operand	Range
RA	Relative Address	X	X represents an index in the range of +127 to –128 which is an offset relative to the address of the next instruction.
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14.
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH.
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH.
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

Table 126 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 126. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow, as shown in the following example.

dst ← dst + src

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

Hex Address: FC5

Table 222. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bit	7	6	5	4	3	2	1	0
Field	PAD7ENL	PAD6ENL	PAD5ENL	PAD4ENL	PAD3ENL	PAD2ENL	PAD1ENL	PAD0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC5H							

Hex Address: FC6

Table 223. Interrupt Request 2 Register (IRQ2)

Bit	7	6	5	4	3	2	1	0
Field	T3I	U1RXI	U1TXI	DMAI	PC3I	PC2I	PC1I	PC0I
RESET	0							
R/W	R/W							
Address	FC6H							

Hex Address: FC7

Table 224. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	7	6	5	4	3	2	1	0
Field	T3ENH	U1RENH	U1TENH	DMAENH	C3ENH	C2ENH	C1ENH	C0ENH
RESET	0							
R/W	R/W							
Address	FC7H							

Hex Address: FC8

Table 225. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	7	6	5	4	3	2	1	0
Field	T3ENL	U1RENL	U1TENL	DMAENL	C3ENL	C2ENL	C1ENL	C0ENL
RESET	0							
R/W	R/W							
Address	FC8H							

Hex Address: FD3

Table 232. Port A–H Output Data Register (P*x*OUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
Address	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

Hex Address: FD4

Table 233. Port A–H GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

Hex Address: FD5

Table 234. Port A–H Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W							
Address	FD1H, FD5H, FD9H, FDDH, FE1H, FE5H, FE9H, FEDH							

Hex Address: FD6

Table 235. Port A–H Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X							
R/W	R							
Address	FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH							

Hex Address: FD7

Table 236. Port A–H Output Data Register (P*x*OUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
Address	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

Hex Address: FD8

Table 237. Port A–H GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

Hex Address: FD9

Table 238. Port A–H Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W							
Address	FD1H, FD5H, FD9H, FDDH, FE1H, FE5H, FE9H, FEDH							

Hex Address: FDA

Table 239. Port A–H Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X							
R/W	R							
Address	FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH							